

AMENDMENTS TO THE SPECIFICATION

Please add the following new paragraph after paragraph [0030]:

[0031] Fig. 2A is a block diagram illustrating a software architecture supporting page cloning according to embodiments of the present invention.

Please replace paragraph [0052] with the following amended paragraph:

[0052] In various embodiments of the present invention, old data cache 240 can be primarily associated with upper-level system 210 or lower-level storage module 220. Alternatively, old data cache 240 can be implemented as a separate caching module, and configured to service requests for the creation of cache pages, writes to/reads from cache pages, the destruction of cache pages, and the like. Thus, in certain embodiments, upper-level system 210 includes old data cache 240, and ~~provide provides~~ access to old data cache 240 to lower-level storage module 220. In other embodiments, old data cache 240 is primarily under the control of lower-level storage module 220, which provides upper-level system 210 with access thereto. In still other embodiments, old data cache 240 is an independent caching module, and so is not under the primary control of upper-level system 210 or lower-level storage module 220.

Please add the following new paragraph after paragraph [0053]:

[0054] Fig. 2A depicts data cache 260. In certain embodiments, a single cache comprises data cache 260 and old data cache 240.

Please replace paragraph [0063] with the following amended paragraph:

[0063] Using a technique according to the present invention, whenever a filesystem dirties a page having valid contents (i.e., contents that are synchronized with the data stored in the storage system), the filesystem can clone the page, copying the existing contents into the new page before modifying the old page. The filesystem also signals

the volume manager, indicating that the volume manager should read and cache the corresponding parity information (parity block(s)). The cloned page is then used by the volume manager in completing the other phases of the write operation being performed. Subsequent modifications to the paged data [[goes]] go to the original page. When this dirty (original) page is flushed by the filesystem, resulting in a volume write, the volume manager can completely eliminate the read phase of the write operation, since the volume manager now has at its disposal the old data (in the cloned page) and old parity (in the cached parity blocks).

Please replace paragraph [0093] with the following amended paragraph:

[0093] Fig. 14 depicts a block diagram of a computer system 1410 suitable for implementing ~~a virtual loader according to~~ the present invention. Computer system 1410 includes a bus 1412 which interconnects major subsystems of computer system 1410, such as a central processor 1416, a system memory 1417 (typically RAM, but which may also include ROM, flash RAM, or the like), an input/output (I/O) controller 1418, an external audio device, such as a speaker system 1420 via an audio output interface 1422, an external device, such as a display screen 1424 via display adapter 1426, serial ports 1428 and 1430, a keyboard 1432 (interfaced with a keyboard controller 1433), a storage interface 1434, a floppy disk unit 1437 operative to receive a floppy disk 1438, a host bus adapter (HBA) interface card 1435A operative to connect with a fibre channel network 1490, a host bus adapter (HBA) interface card 1435B operative to connect to a SCSI bus 1439, and an optical disk drive 1440 operative to receive an optical disk 1442. Also included are a mouse 1446 (or other point-and-click device, coupled to bus 1412 via serial port 1428), a modem 1447 (coupled to bus 1412 via serial port 1430), and a network interface 1448 (coupled directly to bus 1412).